Approximation of an inverse of the incomplete beta function

Mike Giles, Casper Beentjes

Mathematical Institute, University of Oxford

International Congress on Mathematical Software

July 24, 2024

Outline

- motivation
- approximation based on Normal expansion
- approximation based on Gil, Segura, Temme expansion
- future work

Generation of scalar random variables

To generate scalar random variables X with a known Cumulative Distribution Function (CDF)

$$C(x) = \mathbb{P}[X \leq x]$$

one approach is to create a $(0,1)$ uniform random variable, and then apply the inverse CDF

$$X = C^{-1}(U)$$

This works for discrete distributions if $C^{-1}(U)$ is defined appropriately.

Generation of Poisson random variables

Previous work on the efficient generation of Poisson random variables used an inverse of the incomplete gamma function

$$\overline{C}_\lambda^{-1}(u) \;=\; \lfloor C_\lambda^{-1}(u) \rfloor$$

where $\overline{C}_\lambda^{-1}(u)$ is the inverse CDF for the Poisson distribution for rate λ, and $C_\lambda^{-1}(u)$ is the inverse of the incomplete gamma function:

$$C_\lambda(x) = \frac{1}{\Gamma(x)} \int_\lambda^\infty e^{-t}\, t^{x-1}\, \mathrm{d}t.$$

This led to accurate and efficient software on both CPUs and GPUs.

Generation of Binomial random variables

The present work has a similar motivation, to generate Binomial random variables, for which we now have two parameters: n, p.

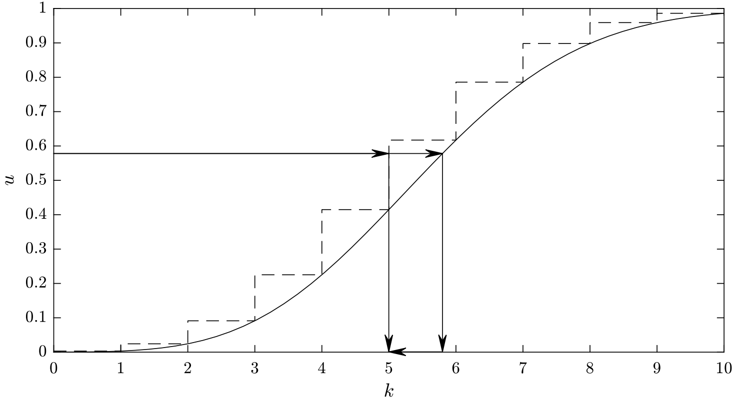
The approach is also the same, using an inverse of the incomplete beta function and then rounding down to the nearest integer:

$$\overline{C}_{n,p}^{-1}(u) \;=\; \lfloor C_{n,p}^{-1}(u) \rfloor$$

where $\overline{C}_{n,p}^{-1}(u)$ is the inverse CDF for the Binomial distribution, and $C_{n,p}^{-1}(u)$ is an inverse of the incomplete gamma function:

$$C_{n,p}(x) \equiv I_{1-p}(n{+}1{-}x, x) = \frac{n!}{(x{-}1)!\,(n{-}x)!} \int_0^{1-p} t^{n-x} (1{-}t)^{x-1}\, \mathrm{d}t$$

Illustration of rounding down procedure



Plot of $\overline{C}(x)$ (dashed line) and $C(x)$ (solid line) for $n = 20$, $p = 0.25$

Generation of Binomial random variables

Two notes:

- We want the inverse of $C_{n,p}(x) \equiv I_{1-p}(n+1-x, x)$ with respect to x; this is different to other inverses for which software exists
- Small errors in approximating $Q(u) \equiv C_{n,p}^{-1}(u)$ can only lead to incorrect rounding for the binomial r.v.'s when near an integer.

  The final software will use a correction process in this case, so we don't need exceptional accuracy – prepared to tradeoff accuracy versus cost

Normal asymptotic approximation

As $n \to \infty$, binomial CDF approaches Normal CDF with mean np and variance npq, where $q = 1 - p$. This motivates a change of variables

$$x = np + \sqrt{npq}\; y, \qquad t = q + \sqrt{pq/n}\,(z-y),$$

with y the deviation from the mean, normalised by the standard deviation.

This leads to

$$C(x) = \frac{1}{\sqrt{2\pi}} \int_{y-\sqrt{nq/p}}^{y} J \,\mathrm{d}z,$$

where

$$\begin{aligned} \log J \;\; = \;\; & \tfrac{1}{2}\log(2\pi) + \log\Gamma(n{+}1) - \log\Gamma(x) - \log\Gamma(n{-}x{+}1) \\ & +(n{-}x)\log t + (x{-}1)\log(1{-}t) + \tfrac{1}{2}\log(pq/n). \end{aligned}$$

Normal asymptotic approximation

An expansion in powers of $n^{-1/2}$, followed by exponentiation and a second expansion in powers of $n^{-1/2}$, yields

$$J(y,z) = \exp(-\tfrac{1}{2}z^2) \left(1 + \sum_{m=1}^{\infty} n^{-m/2} e_m(p,y,z)\right)$$

where $e_m(p,y,z)$ are polynomial in p, y and z. Integrating by parts then gives

$$C(x) = \Phi(y) + \phi(y) \left(\sum_{m=1}^{3} n^{-m/2} \tilde{f}_m + O(n^{-2})\right)$$

where $\Phi(y)$ is the Normal CDF function, $\phi(y) = \Phi'(y)$ is the Normal probability density function, and $\tilde{f}_1, \tilde{f}_2, \tilde{f}_3$ are polynomial in both p and y.

Normal asymptotic approximation

Inverting this expansion, gives the final asymptotic expansion in which $w = \Phi^{-1}(u)$,

$$
\begin{aligned}
Q(u) \;\; = \;\; & np \; + \; \sqrt{npq}\; w \; + \; \left(2 + 2p + (q-p)\, w^2\right)/6 \\
& + \; \left((-2+14pq)\, w + (-1-2pq)\, w^3\right)/(72\sqrt{npq}) \\
& + (p-q)(2+pq)(16-7w^2-3w^4)/\,(1620\, npq) \; + \; O(n^{-3/2}),
\end{aligned}
$$

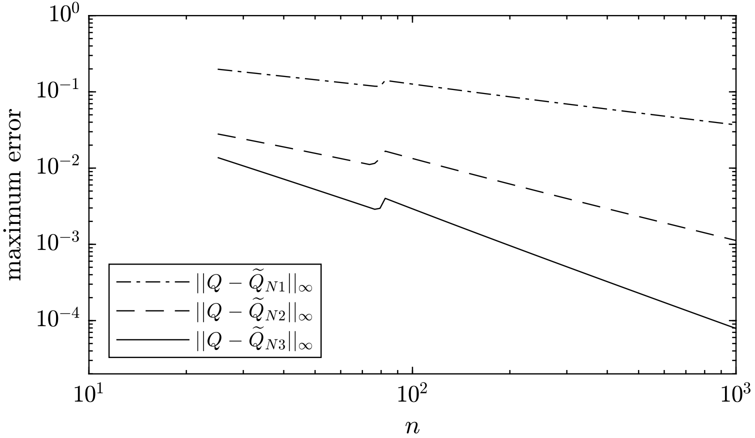
providing the following three approximations:

$$
\begin{aligned}
\widetilde{Q}_{N1}(u) &= np \; + \; \sqrt{npq}\; w + \left(2 + 2p + (q-p)\, w^2\right)/\,6 \\
\widetilde{Q}_{N2}(u) &= \widetilde{Q}_{N0}(u) \; + \; \left((-2+14pq)\, w + (-1-2pq)\, w^3\right)/(72\sqrt{npq}) \\
\widetilde{Q}_{N3}(u) &= \widetilde{Q}_{N1}(u) \; + \; (p-q)(2+pq)(16-7w^2-3w^4)\,/(1620\, npq).
\end{aligned}
$$

The first corresponds to the Cornish-Fisher expansion with skewness correction based on the binomial mean, variance and skew.

Normal asymptotic approximation

Maximum errors for $p = 0.25$



Odd "glitch" is because we limit range to $|w| < 3$ and $10 < x < n-9$; final software will use other methods outside that region

GST expansion

Gil, Segura, Temme (2020) proved that

$$C(x) \approx \Phi(-\eta\sqrt{\nu})$$

where $\nu = n{+}1$, $\xi = x/\nu$ and η is given by

$$\eta = \sqrt{-2\left(\xi \log \frac{p}{\xi} + (1{-}\xi)\log\frac{1{-}p}{1{-}\xi}\right)} \equiv h_p(\xi),$$

Hence, to leading order

$$x = \widetilde{Q}_{T0}(U) \equiv \nu\, \xi_0 \equiv \nu\, h_p^{-1}(\eta_0)$$

where $\eta_0 \equiv -w/\sqrt{\nu}$ with $w = \Phi^{-1}(U)$.

GST expansion

Gil, Segura, Temme (2020) also derived an improved representation

$$C(x) = \Phi(-\eta\sqrt{\nu}) + R_\nu(\eta)$$

with an expansion for $R_\nu(\eta)$.

This leads to an improved approximation

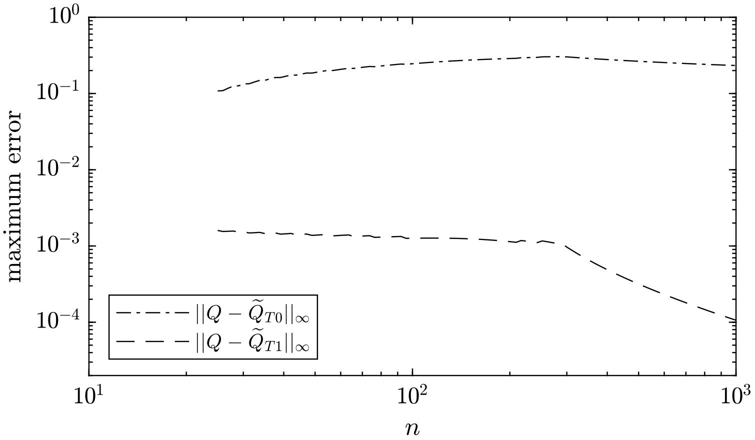
$$\widetilde{Q}_{T1}(u) = \nu\,\xi_0 + g_p(\eta_0),$$

where

$$g_p(\eta_0) = \left\{ \eta_0^{-1} \log\left(\sqrt{\xi_0(1-\xi_0)}\eta_0/(p-\xi_0) \right) \right\} \times \left\{ -\eta_0 \Big/ \left(\log \frac{(1-\xi_0)p}{(1-p)\xi_0} \right) \right\}$$

GST approximation

Maximum errors for $p = 0.25$, $|w| < 10$, $10 < x < \nu - 9$.



GPU software plans

Algorithm for vector implementation (with different n, p, u for each element):

- use "bottom-up" or "top-down" summation (i.e. direct summation to compute $\overline{C}(m)$ or $1-\overline{C}(m)$) when npq is small
- otherwise, construct $\widetilde{Q}_{T1}$ approximation with error bound
  - use "bottom-up" or "top-down" summation when $x<10$ or $x>n-9$
  - if $\widetilde{Q}_{T1}$ is too close to an integer, evaluate $C_{n,p}(x)$ to determine correct rounded value

Note: corrections will be needed very rarely, so excellent vector performance – justifies higher cost of GST approximation

CPU software plans

Algorithm for scalar implementation:

- use "bottom-up" or "top-down" summation when npq is small
- otherwise, define $w = \Phi^{-1}(u)$ and if $|w| < 3$ construct $\widetilde{Q}_{N2}$ approximation, with error bound based on $\widetilde{Q}_{N3} - \widetilde{Q}_{N2}$
  - if $|w| \geq 3$ or if $\widetilde{Q}_{N2}$ is too close to an integer, switch to $\widetilde{Q}_{T1}$ approximation
  - use "bottom-up" or "top-down" summation when $x{<}10$ or $x{>}n{-}9$
  - if necessary evaluate $C_{n,p}(x)$ to determine correct rounded value

Note: reduced cost most of the time, but more corrections needed

References

Gil, A., Segura, J., Temme, N. "Asymptotic inversion of the binomial and negative binomial cumulative distribution functions". Electronic Transactions on Numerical Analysis 52, 270-280 (2020).

Giles, M. "Algorithm 955: Approximation of the inverse Poisson cumulative distribution function". ACM Trans. Math. Software 42(1) (2016)

Temme, N. "Uniform asymptotic expansions of the incomplete gamma functions and the incomplete beta function". Mathematics of Computation 29(132), 1109-1114 (1975)